



## **FERRAMENTAS DE ENSINO DE PLD E VHDL PARA CURSOS DE CURTA DURAÇÃO: UMA PROPOSTA BASEADA EM KITS DIDÁTICOS MODULARES**

**Michel Guterres** – michelguterres@yahoo.com.br  
Instituição de Federal de Santa Catarina  
Avenida Mauro Ramos, 950  
88020-300 – Florianópolis –Santa Catarina

**Francisco Edson Nogueira de Melo** – emelo@ifsc.edu.br  
Instituição de Federal de Santa Catarina  
Avenida Mauro Ramos, 950  
88020-300 – Florianópolis –Santa Catarina

**Adriano Regis** – adriano.regis@ifsc.edu.br  
Instituição de Federal de Santa Catarina  
Avenida Mauro Ramos, 950  
88020-300 – Florianópolis –Santa Catarina

**Roberto Alexandre Dias**– roberto@ifsc.edu.br  
Instituição de Federal de Santa Catarina  
Avenida Mauro Ramos, 950  
88020-300 – Florianópolis – Santa Catarina

**Resumo:** *Esse artigo discute uma proposta de métodos e ferramentas para um curso desenvolvido sobre a implementação da linguagem VHDL em projetos didáticos utilizando kits didáticos e modulares, caracterizando o ambiente integrado de aprendizagem proposto. Apresenta-se também uma proposta de curso com duração de cinco dias, abordando os principais tópicos de programação em VHDL, e demonstrando exemplos didáticos para cada assunto.*

**Palavras-chave:** *VHDL, Curso, Treinamento, VHDL em CPLD e FPGA, Lógica Programável.*

### **1. INTRODUÇÃO**

O VHDL é uma linguagem que descreve a estrutura ou comportamento de um sistema ou componente de hardware, desenvolvida nos anos de 1980 pelo Departamento de Defesa dos Estados Unidos como um padrão para a documentação de dispositivos normatizada pelo IEEE no ano de 1987. Tornando-se de domínio público, a linguagem VHDL se tornou um tópico importante da Engenharia Eletrônica, pois dentre suas vantagens estão: facilidade de atualização de projetos, flexibilidade na escolha de ferramentas e fornecedores, utilização de



simulação para verificar o comportamento do sistema digital e permitir descrever o hardware em diversos níveis de abstração, por exemplo, em algoritmo, comportamental e transferência entre registradores.

No entanto, estando o projeto de circuitos integrados dedicados restrito às empresas de grande porte e provedores de soluções, em um primeiro momento a utilização do VHDL ficou restrita aos provedores de soluções ASIC (Application Specific Integrated Circuit).

Em 1991, Makimoto analisou as tendências da indústria de circuitos integrados, que oscilavam entre soluções padronizadas e personalizáveis em ciclos de 10 anos, e propôs que entre 1997 e 2007 haveria uma demanda pelos “campos programáveis” de dispositivos reconfiguráveis, padronizados na indústria, porém personalizadas na aplicação (MAKIMOTO, 1991). Em concordância com essas previsões surgiram os dispositivos lógicos programáveis PLDs, possuindo como característica principal a capacidade de reconfiguração pelo usuário, eliminando partes do processo de fabricação do chip tradicional e facilitando desta forma as mudanças de projeto.

Como uma evolução tecnológica surge os FPGAs (Field Programmable Gate Array, em português Arranjo de Portas Programável em Campo), que são dispositivos capazes de atender à demanda de projetos grandes e complexos devido à alta densidade lógica e baixo tempo de resposta. Devido às vantagens de uma implementação de lógica em Hardware, os FPGAs podem ser utilizados em diversas áreas de uso dos sistemas embarcados segundo Józmiak, Nedjah e Figueroa (2010), oferecendo maior desempenho, menor consumo e menor tempo de desenvolvimento em relação às soluções baseadas em processadores convencionais ou microcontroladores.

No entanto, atualmente uma das desvantagens da implementação em hardware é a escassez de profissionais que conheçam os conceitos de lógica reconfigurável e dominem suas ferramentas de projeto. De fato, a falta de domínio desse recurso tem restringido a aplicação de lógica reconfigurável apenas às soluções onde a implementação exige um desempenho que não pode ser alcançado por microcontroladores ou processadores.

O currículo básico dos cursos da área de Engenharias IV contemplam eixos curriculares que abordam os conceitos fundamentais de um projeto em nível de hardware como, Lógica Combinacional e Lógica Sequencial. Nesse caso, observa-se que a lacuna está na implementação, ou seja, no domínio das ferramentas (ambientes de desenvolvimento) e linguagens (VHDL, por exemplo) de desenvolvimento de lógica em hardware.

O curso proposto nesse artigo visa promover uma formação básica na implementação de soluções de projeto de hardware em Lógica Programável, com o uso de uma abordagem prática, incluindo diversos projetos didáticos elementares, em um conjunto modular de experimentação (kits didáticos) disponibilizada aos participantes durante o treinamento.

O curso proposto tem uma carga horária de 20 horas-aula, e apresenta aos participantes as principais estruturas da linguagem VHDL, o ambiente de desenvolvimento e algumas de suas ferramentas, além dos kits didáticos e modulares. Para garantir a compreensão dos tópicos tratados, são utilizados exemplos introduzidos no ambiente de desenvolvimento e testados nos kits didáticos.

## **2. KITS DE DESENVOLVIMENTO EM FPGA**

Com o intuito de garantir aos alunos um maior envolvimento com a tecnologia durante o curso, estão disponíveis kits didáticos e modulares desenvolvidos pelo grupo FPGA



PARA TODOS (<http://fpgaparatodos.com.br>), permitindo o desenvolvimento de exercícios práticos em cada tópico teórico. Tais kits possibilitam ao aluno observar o funcionamento dos projetos apresentados durante o curso e, progressivamente, realizar a alteração ou variação de determinados parâmetros de maneira criativa.

Como forma de abranger o maior número de variações de experimentos propostos pelo professor ou pelos próprios alunos, os kits desenvolvidos utilizam um conceito de construção modular. Essa construção modular utiliza módulos principais e módulos periféricos de hardware, fazendo com que o aluno possa variar o circuito elétrico final de seu protótipo através de conexões simples, sem necessidade de soldagem de componentes ou conhecimento profundo sobre os circuitos a serem interligados.

A construção modular permite que os experimentos sejam montados e testados rapidamente e com segurança, garantindo que o aluno dedique a maior parte do tempo nos objetivos principais do curso. A seguir se apresenta uma breve descrição dos recursos já desenvolvidos.

## **2.1.Kits**

Com o intuito de garantir os alunos um maior envolvimento prático com a tecnologia durante o curso, disponibilizar-se-á kits didáticos e modulares que permitem a gravação e verificação do funcionamento dos projetos propostos durante o curso. A construção modular dos kits, baseada em módulos principais e módulos periféricos, permite que experimentos possam ser montados e testados com maior agilidade.

### ***Módulos principais***

Três modelos de módulo principal de lógica programável estão disponíveis para os alunos durante o curso, sendo esses módulos apresentando diferentes níveis de complexidade e capacidade de programação.

Para projetos menores (ou futuros protótipos de baixo curso) utiliza-se a placa com o CPLD EMP7064, um segundo módulo baseado em CPLD disponível pertence à família MAX da Altera, e para projetos avançados é disponibilizado para os alunos um módulo baseado em FPGA da família Cyclone\_IV também da Altera com o dispositivo EP4CE15E22C8. A figura 1 e a tabela 1 apresentam respectivamente, uma foto desses três módulos e suas características técnicas.



Figura 1 – Módulos Principais

Tabela 1 – Comparação Módulos Principais.

Módulo/ Componente	Recurso do Componente	Recursos da Placa		
		Pinos I/O	Clock	Periféricos
CPLD_Max_7000S/ EMP7064SLC44	Número de Macrocelulas: 64 Blocos de matriz lógica: 4 Número máximo de entradas e saídas: 68 Tensão de entrada e saída (V): 3.3 e 5. Entradas globais de clock: 2	26	24MHz	10 frequências de Clock selecionáveis
CPLD_Max_V/ 5M240ZT100	Elementos Lógicos: 240 (equivalente 192 macrocelulas) Memória flash: 8 Kb Tensão de entrada e saída (V): 1.2, 1.5, 1.8, 2.5, 3.3 Entradas globais de clock: 4	48	24MHz	8 leds, led RGB, Buzzer, 6 botões.
FPGA_Cyclone_IV/ EP4CE15E22C8	Elementos Lógicos: 15 Blocos de Memórias M9K: 56 Embedded memory: 504 Multiplicador 18 x 18: 56 Tensão de entrada e saída (V): 1.2, 1.5, 1.8, 2.5, 3.3 Entradas globais de clock: 20	32	50MHz	6 botões, 8 leds.

### ***Módulos Periféricos***

Os módulos periféricos são placas eletrônicas com funções com funções distintas de entrada ou de saída (sensores e atuadores), permitindo uma ampla variedade de experimentos. As placas foram projetadas com o intuito de possibilitar ao aluno, com domínio ainda restrito em eletrônica, montar e propor seus experimentos. Agregando versatilidade ao conjunto

didático, estes módulos periféricos podem ser utilizados em treinamentos de outros sistemas embarcados, como microcontroladores ou interfaceamento com PC via USB.

Um curso introdutório prevê a aplicação dos seguintes módulos periféricos: placa de chaves, placa display de setesegmentos, placa de display de seis algarismos, placa de botões, placa de led e placa leitor ADC.

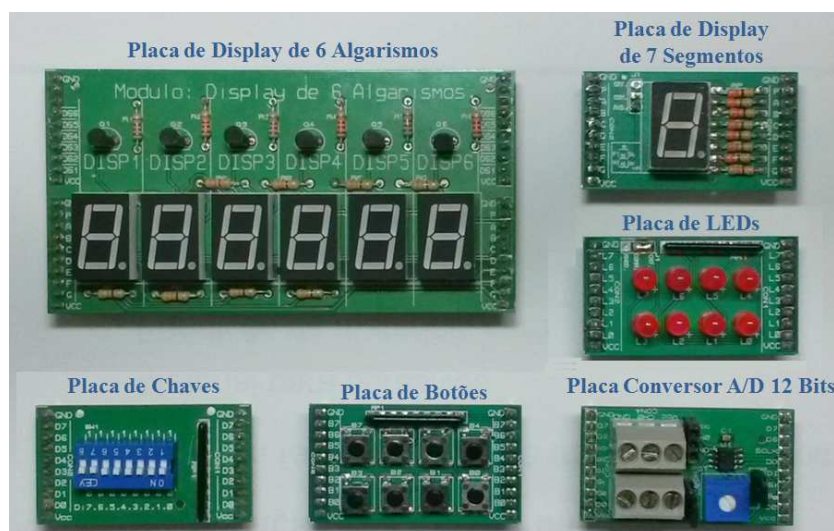


Figura 2 – Módulos Periféricos

## 2.2. Ambiente de Desenvolvimento

Considerando a escolha por tecnologias da fabricante Altera no desenvolvimento dos módulos principais, optou-se também pelo ambiente de desenvolvimento Quartus II (versão 13.1 ou 13.0), cuja versão gratuita encontra-se disponível pra download no site do fabricante (Altera) na data da publicação desse artigo. A versão gratuita permite que os alunos possam continuar seus estudos nas suas casas, em seus próprios computadores, sem o compromisso de adquirir um software de alto custo.

## 2.3. Ambiente Virtual

Todo o material necessário para que os alunos possam acompanhar o curso são disponibilizados em um Ambiente Virtual de Ensino Aprendizagem (AVEA). O acesso ao ambiente é controlado, provendo ao instrutor recurso para verificar a evolução do treinando, ou personalizar o atendimento após a finalização do curso. Os materiais disponíveis no Ambiente Virtual do curso são os seguintes:

### *Apresentações*

Os conceitos teóricos básicos são introduzidos de forma simples, em apresentações multimídia, acompanhados de diversos exemplos práticos. O material didático é desenvolvido em software aberto, e disponibilizado posteriormente a outras instituições como referência de curso para ser livremente ajustado de acordo com suas necessidades.



### ***Projetos***

Diversos exemplos de códigos em VHDL são desenvolvidos ao longo dos cinco dias de curso e testados por simulação nos diferentes módulos didáticos, propiciando ao aluno uma sólida compreensão de conceitos abordados e a oportunidade de começar a sua “biblioteca de modelos” que poderão ser utilizados em projetos futuros, e a experiência com os aspectos práticos de construção de pequenos sistemas digitais funcionais. Estes exemplos estão disponíveis na plataforma AVEA.

### ***Tutoriais***

Tutoriais multimídia são utilizados para mostrar passo a passo o desenvolvimento de programas que apresentam algum novo conceito em VHDL ou uma nova ferramenta do Quartus II.

### ***Fotos e Vídeos***

Todos os programas apresentados nos cinco dias possuem fotos e vídeos do seu funcionamento.

## **3. PROGRAMA DE TREINAMENTO PROPOSTO**

A seguir apresenta-se, a proposta de um curso introdutório (treinamento) a linguagem VHDL e Dispositivos Lógicos Programáveis. O curso foi dividido em cinco encontros, onde o participante é apresentado aos conceitos teóricos juntamente com o desenvolvimento de projetos básicos de lógica programável. Essa estrutura curricular tem sido adotada e melhorada em treinamentos realizados no Instituto Federal de Santa Catarina-IFSC e Universidade Federal de Santa Catarina-UFSC, para alunos de Cursos Superiores de Tecnologia, Cursos Técnicos e de Engenharia.

A apresentação em ordem cronológica permite ao leitor simular e adequar esse treinamento com maior facilidade às suas necessidades, e transpô-lo a seguir para seu plano de aulas ou plano de ensino. A cada dia são apresentados os tópicos principais a serem discutidos no encontro, e cuja ordem de apresentação pretende satisfazer os pré-requisitos necessários subsequentemente.

### ***3.1. Primeiro Dia***

#### ***Arquitetura de dispositivos de lógica programável, conceitos básicos:***

Neste tópico é feita uma explicação de circuitos combinacionais, circuitos sequenciais, dispositivos programáveis (PLD, CPLD, FPGA).

São apresentados os kits didáticos e modulares desenvolvidos pelo grupo FPGA PARA TODOS e que estarão disponíveis para os alunos durante o curso.

#### ***Ambiente de Desenvolvimento:***

O Quartus II é o último ambiente de desenvolvimento da ALTERA para seus dispositivos CPLD e FPGA. Inclui também avançados recursos para o desenvolvimento de sistemas embarcados (“embedded systems”) com suporte a núcleos de processador e DSP.

#### ***Primeiros conceitos em VHDL:***



Neste tópico é desenvolvido o primeiro projeto em VHDL, apresentando a estrutura básica de um código VHDL: declaração de bibliotecas/pacotes, entidade e arquitetura.

Projetos desenvolvidos no primeiro dia:

`e_nibbles` : porta "e" de quatro bits

`somador` : exemplo de soma de inteiros com entradas do tipo inteiro

`soma_e_and`: soma aritmética de quatro bits, e porta "e" de quatro bits, com entradas do tipo `std_logic_vector`.

`teste_atribuicoes`: esse exemplo demonstra a utilização da atribuição de sinais dentro da arquitetura.

### ***3.2.Segundo Dia***

#### ***Processo Implícito:***

Todas as declarações de atribuição implicam em lógica concorrente, operações que executam ao mesmo tempo. A ordem das declarações não implica na modificação do resultado da operação, embora possa resultar menos legibilidade.

#### ***Processo Explícito:***

Projetos que ocorrem sequencialmente, concorrentemente entre si e necessitam de uma lista de sensibilidade.

Lista de sensibilidade define sinais que afetam o resultado de um processo. O processo é executado quando um elemento da lista de sensibilidade muda de valor.

Projetos desenvolvidos no segundo dia:

Decodificador 2 x 4, ULA simples, deslocador de 4 bits, display multiplexado.

### ***3.3.Terceiro Dia***

#### ***Simulação com TestBench***

Um “**Testbench**” é um arquivo de código HDL que incorpora código que se deseja testar, fornece a ele sinais de entrada e processa os resultados de acordo com o desejado.

A ferramenta de simulação executa o código do **Testbench** e retorna os resultados da simulação, em forma de onda e/ou tabular.

Para testar a simulação com o TestBench é utilizado o projeto `int4bits_7seg` desenvolvido no dia anterior.

#### ***Processo Combinacionais***

É um projeto aonde o Processo é sensível a todas as entradas usadas no projeto.

#### ***Processos Sequenciais***

Projeto sensível apenas a um clock e/ou outro sinal de controle. Para o processo sequencial foram desenvolvidos projetos bem simples, que foram testados usando a ferramenta do tópico anterior (TestBench). Os projetos foram: `Flipflop_D`, `Flipflop_D_clear` e o projeto `disp_7seg_reg`.

O projeto `disp_7seg_reg` é uma aplicação simples de um registrador de 4 bits, que permite a exibição de um dado de dois algarismos, em displays de sete segmentos a partir de uma entrada de 4 bits.

Uma placa de chaves manda os dados de 4 bits para o FlipFlop. Esses dados só serão exibidos do segundo display quando ocorrer uma subida do clock.

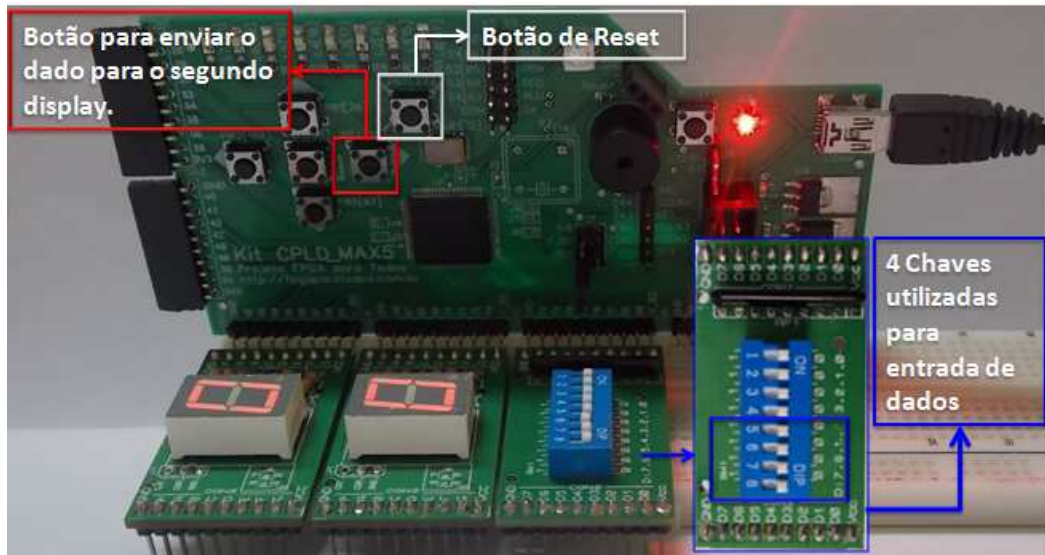


Figura 3 – Projeto disp\_7seg\_reg

#### ***Estudo de blocos de cronometragem***

Neste tópico são desenvolvidas algumas formas de contador em VHDL. Os projetos são os seguintes: Bloco\_clock (utilizado em projetos), contador\_up\_down, contador\_0\_59, base\_cronometragem (utilizado no projeto Relógio estrutural do quarto dia).

#### ***Estudo do registrador serie paralelo***

O projeto Registrador Serie Paralelo pretende aumentar a disponibilidade de pinos de saída em projetos, problemas muitas vezes encontrados.

### ***3.4. Quarto Dia***

#### ***Projeto Hierárquico***

O projeto hierárquico adiciona ao projeto principal, blocos criados a partir de projetos menores. A vantagem de se trabalhar com projetos hierárquicos é a eficiência no desenvolvimento aonde cada integrante do grupo pode desenvolver um componente diferente em arquivos separados, outra vantagem seria que os componentes desenvolvidos podem ser utilizados em projetos maiores, inclusive para outras projetistas.

Para se utilizar esses blocos (funções) desenvolvidos separadamente é necessário **declarar** e **instanciar** esses componentes no projeto principal.

Declaração de componente: Estabelece tipos das portas e os tipos de dados. Para a declaração de componente existe uma ferramenta “InsertTemplate” fornecida pelo Quartus.

Instanciação do componente: Mapeia as portas e um componente de mais baixo nível no projeto aos sinais do projeto corrente



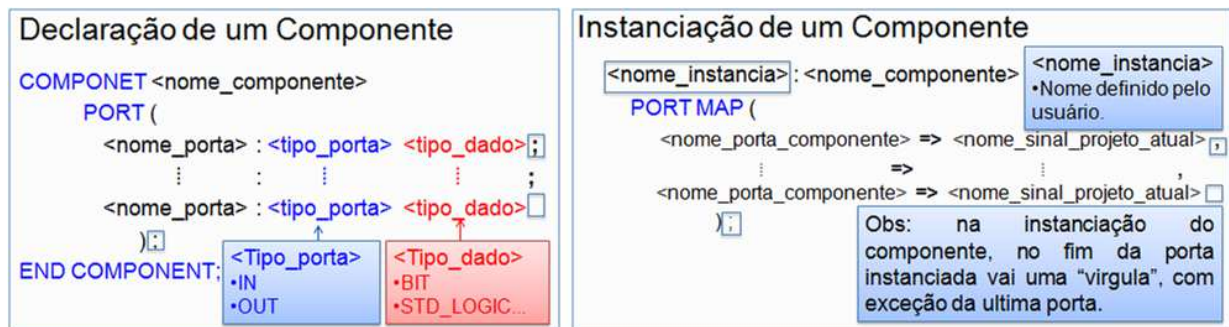


Figura 4 – Sintaxe Instanciação de Componente

O Projeto Hierárquico apresentado no curso é de um relógio estrutural, aonde utiliza projetos menores que foram desenvolvidos nos dias anteriores. Para mostrar as horas é utilizado um display de 6 algarismo.



Figura 5 – Projeto Relógio Estrutural

### ***Megafunções***

A empresa Altera fornece uma série de funções já prontas, que podem ser modificadas conforme a necessidade do projeto. Para ter acesso a essas Funções e modificá-las basta acessar o assistente **MegaWizard Plug-in Manager** e declarar a biblioteca Imp no início do projeto.

As vantagens da utilização das Megafuncoes são: Padrão industrial, grande blocos de lógicas já prontas, fácil reconfiguração.

### ***Implementação de memória em VHDL***

Utilizando o conceito de Megafunções e utilizando o assistente MegaWizard Plug-in Manager é desenvolvido uma memória que ira armazenar 8 bits em 4 endereços diferentes.

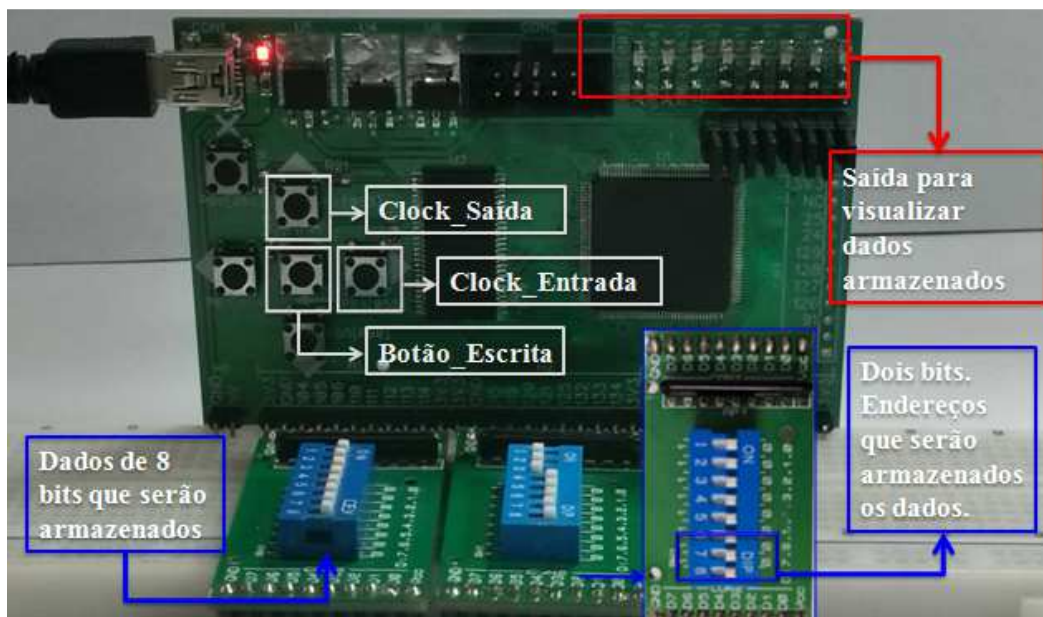


Figura 6 – Projeto Mega Memória

### 3.5. Quinto Dia

#### Máquina de estado

Para o quinto dia o tema abordado foi Máquina de estado, os projetos desenvolvidos foi o acionamento de um motor de passo utilizando Máquina de estado, e uma máquina de estado para fazer a leitura de um AD/C de 12 bits se comunicando via SPI e mostrando o valor lido (valor da tensão em hexadecimal) em um display de 6 algarismos.



Figura 7- Máquina estado Le\_ADC\_SPI



#### 4. CONCLUSÃO E TRABALHOS FUTUROS

Considerando os objetivos do curso em capacitar os participantes em lógica reconfigurável, cujo público alvo é composto por estudantes de tecnologia e engenharia em seus diferentes níveis, esse treinamento atende, de forma complementar, as competências e habilidades desenvolvidas durante o curso regular.

A abordagem baseada em experimentos práticos, e kits de desenvolvimento modulares, permite que o aluno domine e proponha soluções para suas necessidades em um tempo reduzido. De fato, essa abordagem permite que alunos vislumbrem a aplicação de Lógica Reconfigurável em seus projetos (TCCs, Dissertações).

Com poucas alterações, esse treinamento pode ser adaptado para uma capacitação docente para professores que buscam adquirir alguma prática de experimentação com componentes básicos de lógica programável.

Em uma evolução do programa do curso pretende-se explorar novos tópicos no quinto e último, incluindo a análise de temporização e sobre as vantagens da utilização da biblioteca numérica em relação às outras bibliotecas que envolvem operações.

Como trabalhos futuros, sugere-se uma ampla discussão dos resultados da aplicação de treinamentos de lógica reconfigurável realizados em diferentes níveis e cursos, apontando indicadores de desempenho e que realmente e proponham melhorias para os treinamentos seguintes.

##### *Agradecimentos*

Os autores desse trabalho agradecem, em especial aos colegas do Projeto FPGA PARA TODOS onde foram desenvolvidos os Kits Didáticos e o material de orientação apresentado nesse artigo, bem como ao Grupo de Sistemas Embarcados e Distribuídos-NERsD onde esse projeto foi desenvolvido juntamente com as demais pesquisas dos professores e bolsistas, permitindo que as atividades sinérgicas e recursos compartilhados do grupo auxiliassem no desenvolvimento das atividades.

Por fim, ao CNPQ e ao IFSC pela concessão das bolsas que permitiu ao estudante participar desse projeto e ingressar em suas primeiras pesquisas acadêmicas

#### 5. REFERÊNCIAS / CITAÇÕES

ALTERA. **Altera Product Catalog.** Disponível em: <<http://www.altera.com/literature/sg/product-catalog.pdf>> Acesso em: 12 junho. 2014.

ALTERA. **Download.** Disponível em: <<https://www.altera.com/download/sw/dnl-sw-index.jsp>> Acesso em: 11 junho. 2014.

COSTA, Cesar de. Elementos de lógica programável com VHDL e DSP, Teoria e Prática: São Paulo, Érica, 2011. 296 p.

ERCEGOVAC, Milos. Introdução aos sistemas digitais. Porto Alegre: Bookman, 2000. 453 p.



FPGA PARA TODOS. Exemplos com CPLP. Disponível em: <<http://fpgaparatodos.com.br/exemplos-de-aplicacao/exemplos-didaticos.html>> Acesso em: 24 maio. 2014.

JOZWIAK, Lech; NEDJAH, Nadia; FIGUEROA, Miguel. Modern development methods and tools for embedded reconfigurable systems: A survey. Integration: the VLSI journal, Philadelphia, p. 1-33. 25 jun. 2009.

MARCON, César Augusto Missio et al. Modelagem e Descrição de Sistemas Computacionais - Um Estudo de Caso de Comparação das Linguagens VHDL e SDL. In: Iberchip, 2002, Guadalajara, Mexico. VIII Work Shop Iberchip. Los Alamitos: IEEE computer society, 2002. v. 1. p. 1-5.

MAKIMOTO, Tsugio. The Hot Decade of Field Programmable Technologies. Sony Corporation, Tokyo, n. , p.1-4, 01 jan. 1991.

PREDONI, Volnei. Eletrônica digital moderna com VHDL. Rio de Janeiro: Elsevier, 2010. 619 p.

STEINBACH, R. ; MELO, F. E. N. ; DIAS, R. A. . Fpga Para Todos: Um Projeto Para a Disseminação da Tecnologia de Lógica Programável. In: COBENGE - Congresso Brasileiro de Ensino de Engenharia, 2011, Blumenau /SC. XXXIX Congresso Brasileiro de Educação em Engenharia - COBENGE 2011, 2011. v. 1.

## **TOOLS FOR TEACHING AND PLD VHDL FOR SHORT COURSES: A PROPOSAL BASED ON MODULAR TEACHING KITS**

**Abstract:** *This article discusses a proposal for methods and tools developed for a course on the implementation of the VHDL language learning and teaching projects using modular kits, featuring the proposed integrated learning environment. It also presents a proposal for a course lasting five days, covering major topics in VHDL programming, teaching and demonstrating examples for each subject.*

**Key-words:** *VHDL, Course, Training, VHDL for CPLD and FPGA programmable logic.*